NEW APPROACH TO DES WITH ENHANCED KEY MANAGEMENT AND ENCRYPTION/DECRYPTION SYSTEM

(DES ULTIMATE)

[1]Arvind Kumar Sharma, [2]Hitesh Sharma
Department of Computer Science and Engineering
Lovely Professional University
Phagwara (Punjab), India

ABSTRACT
Encryption Algorithms play a very important role in the area of Networking. There are three major issues which any algorithms to deal with, which are Confidentiality, Integrity and, Availability. Some algorithms are for managing Confidentiality other for managing the Integrity. Different level of security provided by different algorithms depending on how difficult is to break them. The most well-known algorithms are DES, AES, 3DES, Blow-fish etc. Proposed algorithm is the enhanced version of DES which based on strong key management and encryption/decryption process and this is for the third party authentication such as Kerberos or Radius. We are using the S-boxes which is used in AES algorithms in-spite of 8/32 S-boxes in order to provide more security because 8/32 S-boxes are more vulnerable to attacks. Self-Invertible matrices also used in specified rounds. We are reducing the computation time for whole process as compare to Triple DES.

KEYWORDS-*Algorithms; Encryption; Decryption; Cipher; Confidentiality; Integity; Authentication; Server; Invertible-Matrix;*

I. INTRODUCTION

Cryptography is known as the study of secrets. This is basically connected to the definition of providing security with encryption/decryption process. Encryption is the process through which the actual plaintext information or message converted to new un-understandable message with the help of encryption algorithms in order to hide the actual meaning of message which is to be stored or transmitted over the channels. And decryption is the reverse of encryption process. For both processes same secret key to be used that is to feed to algorithms for performing such tasks. Security mechanism require specific algorithm for encryption/decryption purpose, and for managing the sub-keys that are to be used to make cipher text from standard plaintext. As the security of the algorithms directly related to key length of secret key, longer the key stronger the technique will be but with longer key computation power of CPU must be affected.
An algorithm will be stronger if it is difficult to recover the plaintext if we have substantial amount of ciphertext available and, complex structure which in case of one key system for managing the bits patterns of the actual data for hiding the relationship of one pattern with other whether it belongs to key or actual message.

A. Way how to convert plaintext:

In stream cipher character by character conversion takes place with particular key and in block cipher a defined set of elements or bits called a block converted at a time to ciphertext with key and then next block feed to the algorithm for conversion this process continuous until whole of the message converted to cipher-text.

B. Type of Operation used for conversion of plaintext to ciphertext:

There are two basic principles that are to be used for plaintext to ciphertext conversion which are:
Transposition: Elements are reorder or rearranged.
Substitution: Every element mapped to another elements.

C. Number of Key Used:

In system where same key to be used by sender and receiver than it is referred to as symmetric key encryption and when both sender and receiver use different key then it is referred to as asymmetric key encryption. In case of symmetric key cryptosystem same key shared by two end which are two client systems or client and server system or two servers to each other for performing encryption-decryption process on the sending plain-text, receiving cipher-text message. But in case of asymmetric cryptosystem during encryption process sender used receiver's public key to perform encryption process and receiver use its own private key to perform decryption process. Both users private key are known only to them but their public key know to everyone who wants to interact with each other.

The rest of the paper is organized as follow. Section II describe the original DES & Triple DES algorithm, Section III describe the Proposed algorithms, Section IV give a comparison of such previously existing algorithms with proposed algorithms based on time taken by algorithms to computer key management and encryption decryption process with the help of suitable diagrams and, Section V provide conclusion and my future work on these type of cryptosystems with some new techniques. And at end acknowledgement, References takes place.

II. DATA ENCRYPTION AND TRIPLE DATA ENCRYPTION STANDARDS

A. Data Encryption Standards (DES)

The DES [11][12][13] was published by the United States' National Bureau of Standards in January 1977 as an algorithm to be used for unclassified data. Data Encryption Standards was one of the strong and popular algorithms used for converting actual plain-text message to cipher-text message. In this technique the whole process works on set of 64-bits at a time with 56-bits of key, actual key is 64-bits long but every eight bit removed from this set and 56-bits are used for further key management.

DES operates on Feistel Network [10] for which uses Confusion-Diffusion process for substituting the bits and permutation for rearranging the bits location. At first Initial Permutation IP rearrange 64-bits. Then In each round of total of 16 rounds at first these 64-bits split into two 32-bits block is called as Left Block L_i and other is called Right Block R_i according to Feistel structure right 32-bits expanded to 48-bits and Xor operation applied on it with the help of 16 sub-keys for each round which is also 48-bits key long and, then with standard 8 S-boxes 48-bits result converted into 32-bits, these 32 bits are again permuted and resultant 32-bits are Xor with left 32-bits, after this the result stored in right R_{i+1} and L_{i+1} contain the previous R_i. This process work for each round of total of 16 rounds and at the end of 16 round final swap occurs and combined 64-bits go through Inverse of Initial Permutation IP^{-1}. This is the cipher text of 64-bits long.

In case of decryption whole process is same just use cipher-text's 64-bits in place of plain-text 64-bits and used 16 sub-keys in reverse order starting from 16 to 1 for rounds 1 to 16.

B. Triple Data Encryption Standards (3DES)

Triple DES is the modification of Standard DES algorithms for providing highly secure data as compared to DES with three different keys of total 168-bits key. That means sixteen rounds on particular 64 bits performed three times with three different keys and their sub-keys so as to make algorithms beyond the reach of brute-force attack performed by EFF DES cracker [7]. In Triple-Des with first key K1 encryption performed on 64-bits plain-text message and then with key K2 decryption performed on the output made by K1 and then at the end with key K3 again encryption performed on the output made by key K2. The result is cipher-text. In case of decryption same process is used but in reverse order here process starts with key K3 to perform decryption, encryption with K2 and at the end decryption with the key K1. The result will be original plain-text.

Although triple DES much secure than standard DES it consume three times CPU power than DES [8][9][12]. In forms of triple DES, algorithm is considered to be more secure, in spite of having

theoretical attacks. But because of similar type of key management technique use by this scheme so brute-force is very straight-forward process when applying with the help of multiprocessor system in order to crack the key, because there is no dependency of key to each other for managing the internal process.

Pseudo Code: Data Encryption Standards
Input: Plain-Text P = P1, P2….P64; Key K = K1, K2….K64 (Include 8 parity bits)

Step1: (Key-Schedule) Compute 16 sub-keys of 48-bit length from K for 16 rounds.
Step 2: (L_0R_0) = IP (P1, P2….P64) by using IP table permute 64-bits; Then split the 64-bits into Left L_i and, Right Ri 32-bits halves;
Step 3: (16 Rounds) such that (i = 0 to 15), compute Li and R_i as follow:
(3a): $L_i = R_{i-1}$
(3b): $R_i = L_{i-1}$ XOR f (R_{i-1}, K_i)
where f (R_{i-1}, K_i) = P(S (E (R_{i-1} XOR K_i))) calculated as follow:
(a) Expand 32-bits R_i to 48-bits and XOR with K_i
(b) Use S-boxes to made 32-bits from 48-bits of (a)
(c) Permute 32-bits returned from (b)
Step4: Swap final block $L_{16}R_{16} = R_{16}L_{16}$
Step5: Inverse Permutation on collective 64-bits returned by (Step4) $IP^{-1}(L_{16}R_{16})$

Output: Cipher-Text C = C1, C2….C64;
where C = ($IP^{-1}(L_{16}R_{16})$)

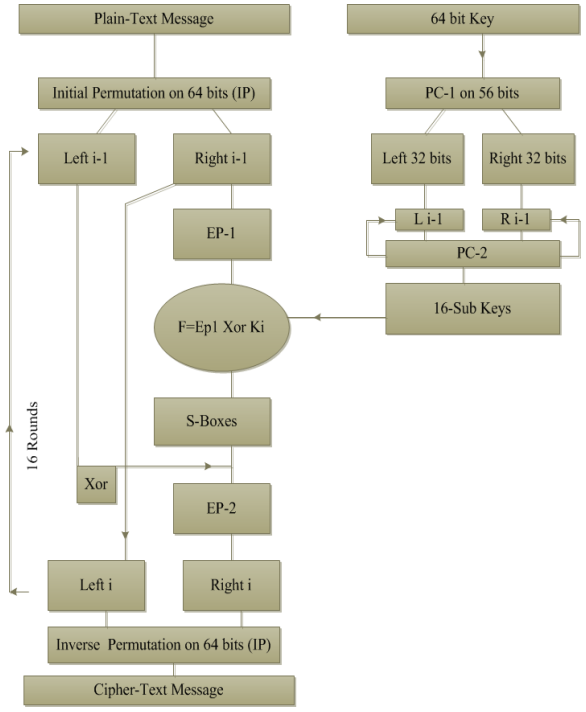

Fig 1: Data Encryption Standard (DES)

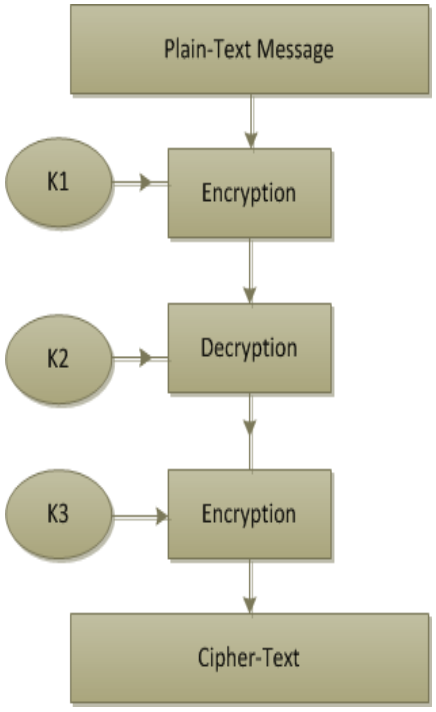

Fig 2: Triple-Data Encryption Standard

III. PROPOSED ALGORITHM OF DES

This research proposed new improvement to DES with enhanced key management system with the help of another two keys H, I which are made from 256-bits Key (K-256) this is our second parent key, which works collectively with actual 64-bits key which is our first parent key (K-64) from this key actual 16-sub keys are made. And there are some set of mathematical computation according to which shift operations performed on key's bits in order to make dependency of one key on to the other key and the calculation of keys which are actually take part in encryption and decryption

operations by working under 16 rounds for increasing the complexity of identifying the actual order of key and the message bits by cryptanalysis and to identify the relationship between keys and message.

Key "K" Management: For the management of Key K (K-64) at first we reduce it to 56-bits by removing parity bits then find the Locator1 and Locator2. Locator1 is number of 1's in 56-bit key and Locator2 is number of 0's in the same 56-bit key Locator1, Locator2 updated with Shift1 Permutation and then Num1 which is Summation (Lacator1, Locator2) calculated and, then with S-Boxes (AES S-Box) again compute the substituted 56-bits and, then apply on these 56-bits Left-Shift according to Prime number P1 and then Right-Shift with Prime number P2 to the resultant 56-bits after applying Left-Shift these prime numbers are made from the key H, which are then split into two halves of 28-bits long KL_i and KR_i from which 16 sub-key made with the help of predefined Schedule of Left-shift which is used in original DES.

Key "H" Management: In this process for calculation of Key H at first enter 256-bits key and sub-divide 256-bits into 8 32-bits pairs then apply Right-shift with Locator1 on every even pair from 8 pair of 32-bits then on resultant 8 pair of 32-bits apply XOR operation by taking 2 pair of 32-bits at first and then XOR result with next 32-bits taking one pair from remaining pairs until all 8 pairs used, at the end we get single 32-bits which is the H0, on to H0 Left-shift operation applied with Locator2 then we get H1. And then XOR is applied on H0 with H1. Resultant is the H2 of 32-bits and then from this H2 make 16-decimal numbers by combining every 2-bit into one decimal number denoted as D_i this is our H key, which we used in 16 rounds.

From these 16 decimal numbers D_i we calculate Num2 which calculated as $\sum \mathbf{D_i}^{(\mathbf{D_j})}$ where **"i"** is odd number and **"j"** is even number up to 1to16. From Num1 and Num2 we calculate Num3 which is equal to Multiplication (Num1, Num2). Prime Number P1 is the first Prime number to the Left of Num3 and P2 is the first Prime number to the right of Num3.

Key "I" Management: In the process for concluding Key I which is the key from which we calculate one Self-Invertible 4*4 matrix (SIM) [2][5][6], which used just in Round No. (1) & Round No. (16), from the total of 16 rounds during Encryption-Decryption Process for making rearrangement of bits of Li before sending to # function during Encryption and after # Function during Decryption. For calculation of I which is 2*2 matrix we take Num1, Num3, P1, P2 and apply Mod16 on these four values which are then updated with corresponding result of Mod16 operation and used for making 2*2 matrix from which the actual calculation of Self-Invertible matrix take place.

With this Self-Invertible matrix (SIM) we encrypt the Li by making at first 8 decimal numbers from 32-bits of Li by taking 4-bits at a time to make a decimal number with in the range of 0-15. Then encrypt 8 decimal numbers with SIM by making matrix multiplication on 4 decimal numbers at time then next 4 decimal numbers with same SIM. After encryption we got 8 new decimal numbers from which again 32-bits made which are then used for # function processing. In case of Decryption same process applied on 32-bits which comes out after # function on specified rounds with same SIM to get the original 32-bits which we encrypt during specified rounds of Encryption process. This whole process with SIM is called Function "A". Function "A" is used before # function during encryption and Li is the input to "A" and after # function during decryption and output of # function is input to "A".

"# Function" Management: In order to increase security we are also using # Function [1] related process. With the help three inputs such as, three pair of 16-decimal number made from (Li and Ri) by taking two bits at a time and made from these two bits a decimal number in the range of (0-3) until 32-bits processed for (Li and Ri) individually and original H which is 16-decimal number. From these three inputs truth-tables read.

Pseudo Code: DES Ultimate Algorithm
Input: Plaint-Text P = P1, P2…..P64; Key K-64 = k1, k2….k64 (Including 8-parity bits); Key K-256 = K1, K2….K256;

Step 1: (Key Management)
Compute from Key K-64 16-sub keys, Key H from K-256 and, Key I. The process is as under below:

a) Make 56-bits key from K-64 by applying PC-1; then calculate Number of 1's and 0's in the resultant 56-bits and, store result in two temporary integer variables says Locator1 (for 1's), Locator2 (for 0's) and apply mode32 on Locator1, Locator2 then result is the location of [Table: Shift1], update the Locator1 and Locator2 with corresponding value in the table.
b) By S-Box substitute 56-bits of (a) into new 56-bits.
c) Key H Manipulation:
    1. Subdivide K-256 into 8 pairs of 32-bits (P1,P2,P3,P4,P5,P6,P7,P8); Apply right-shift on 4 even 32-bits pairs (Pair No: P2,P4,P6,P8) with the help of Locator1;
    2. Apply XOR on total of 8-pairs of 32-bits using updated even pair from (1) and previous odd pairs (Pair No: P1,P3,P5,P7) of 32-bits; such as:
    H0 = ((((((P1xorP2) xor (P3)) xor (P4)) xor (P5)) xor (P6)) xor (P7)) xor (P8)
    3. Apply left-shift with Locator2 on H0;
    H1=left-shift (H0) with Locator2;
    4. H2 = H0 xor H1;
    5. "H" is 16-decimal numbers made from H2 by taking 2-bits at a time from total of 32-bits;
d) Calculate Num1, Num2, Num3 and two Prime Numbers P1, P2 as under below:
    1. Num1 = Locator1 + Locator2;
    2. Num2 = $\sum \mathbf{D_i}^{(D_j)}$ where i is odd decimal number and j is even decimal number from H. where i=1,j=2; i & j<=16;
    3. Num3 = Num1 * Num2;
    4. P1 is first left-most Prime number of Num3 and, P2 is first right-most prime number of Num3;
    5. Update above calculate variables:
    Num1 = (Num1) mod16;
    Num3 = (Num3) mod16;
    P1 = (P1) mod16;
    P2 = (P2) mod 16;
e) Key I Manipulation:
    1. "I" is a 2*2 Matrix having values,
    $A_{22}[0, 0]$ = Num1; $A_{22}[0, 1]$ = Num3; $A_{22}[1, 0]$ = P1; $A_{22}[1, 1]$ = P2; This "I" is used for making (SIM) a 4*4 Self-Invertible Matrix;
    2. Then from this I calculate A11, A12, A21 of self-invertible matrix (SIM);
f) Then with P1, P2 calculated in (d) first apply left-shift with P1 onto 56-bits calculated in (b) and then apply right-shift on result of left-shift with P2 such as:
Right-shift with P2 (Left-shift with P1 (56-bits))

g) 16-Sub Key Manipulation:
Make 16-sub keys with standard DES functions from 56-bits calculated in (f), named as
k1, k2,….k48

Step2: (Encryption Process)

a) Initial Permutation (IP) on 64-bits of Plain-text P1, P2….P64;
b) L0 = P1, P2…P32; R0 = P33, P34….P64;
c) Round i = 0 to 15 repeat
   1. Make 48-bits from 32-bits of right part EP1(Ri);
   2. F = EP1(Ri) xor ki
   3. (And-Xor function [1]) with 6 S-boxes work as:
      i. Subdivide 48-bits output of F into 6 pair of 8-bits;
      ii. Append every 8-bits with 24 0's from start to make 32-bits 6 new pairs named as B1, B2, B3, B4, B5, B6;
      iii. Then read from S-box every entry as: Si(Bi), update Bi; where i=1 to 6;
      iv. Then on Output from iii. Apply the following process: ((((B1 AND B2) xor (B3)) AND (B4)) xor (B5)) AND (B6); we have 32-its output as a result;
d) Apply EP2 on the result of (c,(iv));
e) Function A:
   If (i==0 || i== 15) then
   Encrypt (Li) with SIM;
   (By Making 8 decimal number from 32-bits of Li, after encryption makes 32-bits again from 8 decimal numbers and pass updated Li to #)
   Else
   Li Remain unchanged; (pass to #)
f) Compute # function with Li, (d), H;
   (By making 3 pair of 16-decimal numbers of all of three to read 4 pre-defined Truth-Tables to get 32-bits result and pass it to (Ri+1).
g) Li+1 = Ri; Ri+1 = (f); **(End Loop (c));**
h) Final Swap of Li, Ri;

Output: Cipher-Text C = C1, C2….C64; C = IP^{-1} (Li,Ri);

Algorithm 2: DES Ultimate Algorithm

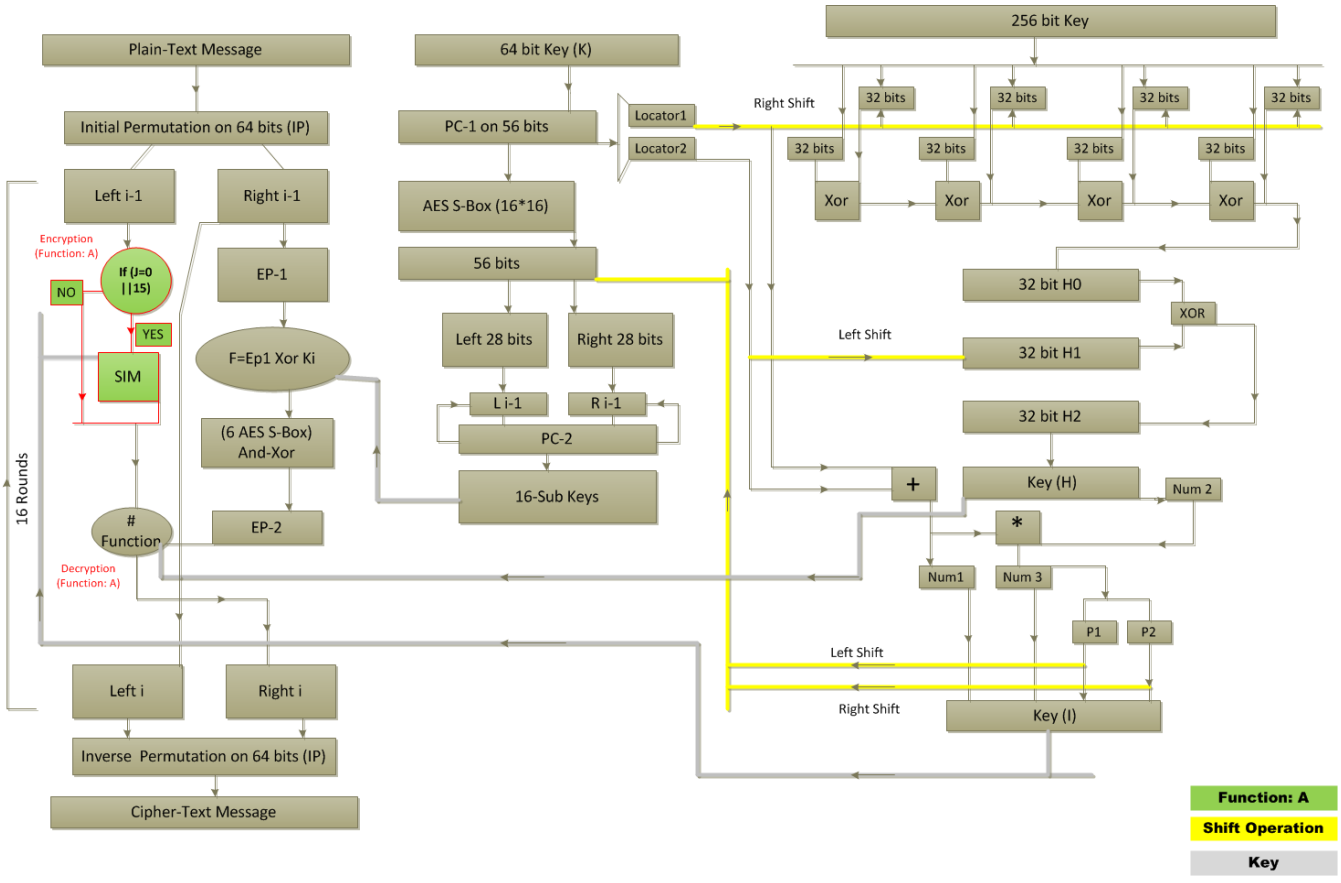

Fig 3: DES Ultimate Flowchart

Encryption/Decryption Process: Consequently, multiple keys are used in each round during encryption-decryption process which is applied on actual plain-text and, cipher-text. During encryption at function (f) 16 sub-keys called ki are used to perform xor operation. Then Key H is used in "#" function to read truth-tables collectively with output of EP-2 and output of function "A". Function "A" make manipulation on 32-bits of Li only during round number 0 and 15 in else case it produce same 32-bits of Li. This same process is used in all 16 rounds of algorithm for total of 64-bits at a time.

In decryption operation which is applied on cipher-text same process is used which is used for encryption operation on plain-text of 64-bits. Just one modification takes place during decryption which is uses of function "A". This function is used after "#" function operation in decryption with same self-invertible matrix (SIM) which is made from Key I. The reason of using it after "#" function is because encrypted input is forward to "#" function in specified round so when output comes out from "#" function then it needs to be decrypted first then forward to Ri. If we use it before "#" function as we do in main encryption process then encryption operation performed not the decryption with same (SIM) on upcoming bits in main decryption process.

[31, 29, 23, 19, 17, 13, 11, 7, 5, 3, 2, 30, 28, 26, 24, 22,
20, 18, 16, 14, 12, 10, 8, 6, 4, 27, 25, 21, 15, 9, 1, 0]

Table 1: Shift1- Permutation

Now, here using this proposed algorithm solve example. The implementation of this algorithm is done in C#.Net. Our input message is simple plain-text message which is first converted into hexa-decimal format for each character of actual message then from this hexa-decimal value we made a binary format of the whole message. With both the keys K-64, K-256 we made key's as (ki, H, I) with the help of these keys we use 16-rounds operation on plain-text and convert it into specified cipher-text during encryption, similarly we convert cipher-text into actual plain-text during decryption.

Step1: Create the Key-64, H, I
K64 Hex-Decimal: 4445534B65793634
K64 Binary:
0100010001000101010100110100101101100101011110010011011000110100
Key H:
1000200313312112
Key I:

12	08
05	09

Step2: Create Plain-text
Plain-Text (P): **ULTIMATe**
Hexa (P): 554C54494D415465
Binary (P): 0101010101001100010101000100100101001101010000010101010001100101
Step3: Initial Permutation on Input
IP: 1111111101000101110101111011100100000000100000000000 1101000000000
Step 4: i=0 to 15 rounds
L0: 11111111010001011101011110111001
R0: 00000000100000000001101000000000

L (Final): 10111110001111000101101011000100
R (Final): 11000011011010111011100111110110
Step5: Inverse of Initial Permutation
IP^{-1}: 10101000111001100101001101111100010111100111101010100111111001011
is: ¨**æS|^z§Ë**.

IV. COMPARISON WITH EXISTING ALGORITHMS

The proposed Algorithm works with the help of three keys ki which is collection of 16-subkeys, each 48-bits of length which is made with the help of K-64 a 64-bits long first Parent key and key H, I which are made with the help of second parent key K-256 a 256-bits of length. H is 32-bits long from which 16 decimal numbers are made and I used to make Self-Invertible matrix which is used in specified rounds from total of 16 rounds during encryption-decryption process.

Now as said proposed algorithm Des-Ultimate use three keys and our comparative Triple-Des also uses three keys and Des just using one key of 64-bits of length. Triple Des keys are ka, kb and kc which are collection of 16-subkey also 48-bits of length made with the help of three different parent keys Ka-64,Kb-64 and Kc-64 each parent key is also 64-bits of length. Both the algorithms using three keys but Triple-Des just follow similar method to make keys from parents keys, and also during encryption-decryption process same process is performed three time and there no strong key management system used and, dependency of keys to each other in order to make crypt-analysis type of work stronger for intruder to get the actual pairs is not available in triple-des. And also in triple-des brute-force is straight-forward but in our proposed work brute-force not possible as all the keys are dependent on each other each key have information to manage the other key so to identify the pairs not possible.

Also we have some functions as function "#", function "A" used in between the processing of 16-rounds so the shuffling of bits from original position to other and encryption of bits by using their corresponding decimal values by SIM make the analysis more complex and, relationship between keys to cipher-text and outputs of every-round with each other not disclosed anymore. We are also using a new defined permutation Shift1 of 32 values which is used to substitute the values of Locator1, Locator2. The actual value of Locator1, Locator2 parameter is first mode with 32 and corresponding value read from the Permutation table Shift1.

From the computation of both the algorithms we conclude that Triple-Des taking more time to encrypt-decrypt the plaint-text of various lengths of character strings as compare to our Des-Ultimate on to the same set of character strings of the plain-text and also for making three keys in both the algorithms also Triple-Des taking more time as compare to proposed algorithm as shown by the figure below:

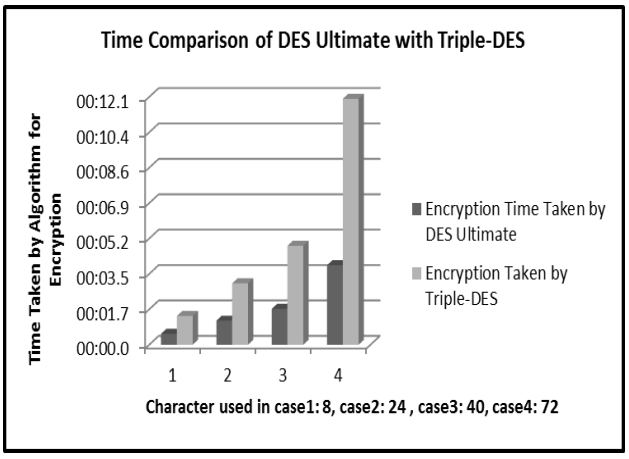

Fig 4: Encrytion Computation time comparison

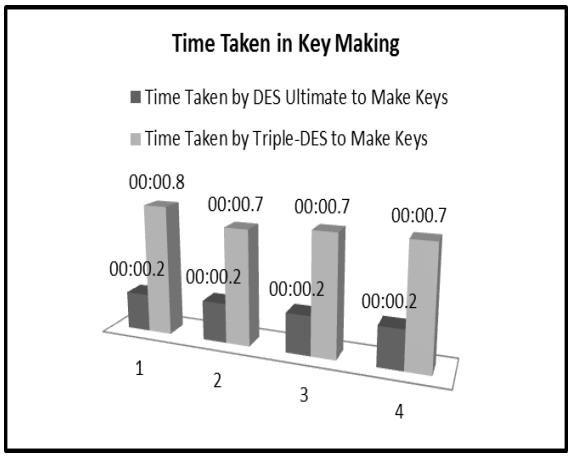

Fig 5: Key Computation time comparison

Apart from time based comparison of Des-Ultimate with Tripe-Des, the proposed algorithms also much stronger than both Triple-Des and Des as we are using strong key management system and some new functions which increase the security of produced result against various attacks as compare to Triple-Des and Des which are just following the same process for encryption-Decryption and we are using AES's S-boxes as compare to S-Boxes which are used by Triple-Des and Des, because S-Boxes are more vulnerable to attack so instead of using 8/32 S-boxes usage of 16*16 S-box is better

option with the help of And-Xor operation to reduce the security related risks from which s-boxes affected.

V. CONCLUSION AND FUTURE WORK

As we are in the society where automated information processing resources are increased day by day and cryptography will increasingly showing its importance as a security mechanism. All the network banking, ecommerce operations, information storage and capture, government applications, online trading through ecommerce websites will need improved and strong method to provide data security. There are various cryptographic algorithms which are providing security to these resources and to data. But Des is less secure and Triple-Des using huge amount of time to perform operations and there some analytical results which demonstrate theoretical weakness of these systems corresponding to generated cipher's. So it is quite important to include some new level of security to these types of algorithms so they will be applicable to provide much security. By new key management system and usage of new functions and replacing the older s-boxes related working and using new technique with s-boxes in place of just xor operation results most reliable, robust and less time consuming DES algorithm Des-Ultimate and make it stronger against any type of attack and intruding. Des processing with new type three keys in-spite of just one key as in original Des and similar three keys in Triple-Des already increases the efficiency of cryptosystem. In future I want to enhance the security of this system and other crypto-systems with the help of image base key management. Because image based protection perform quite well if system capture images from large database on random manner and then capture some binary representation from that selected image which used further in key management system.

ACKNOWLEDGEMENT

First I would like to thank to Department of Computer Science & Engineering, Lovely Professional University, Phagwara which was always there for us listen our problems, give there valuable advice and provide resources for research. I also want to thanks to respected Asst. Prof Md. Attaullah Khan Sir for their support during my dissertation and, to my guide Asst. Prof Hitesh Sharma. I am also grateful to the reviewer for fruitful comments. Thanks to my friends and colleague who have been a source of inspiration and motivation that helped to me during my dissertation period. And all the other people who directly or indirectly supported and help me to fulfill my task. Finally, I heartily appreciate my family members for their motivation, love and support in my goal.

REFERENCES

[1] Shah Kruti R., Bhavika Gambhava, "New Approach of Data Encryption Standard Algorithms" In the International Journal of Soft Computing and Engineering, Volume-2, Issue-1, March 2012.

[2] Bibhudendra Acharya, Girija Sankar Rath, Sarat Kumar Patra, "Novel Methods of Generating Self-Invertible Matrix for Hill Cipher Algorithm" In the International Journal of Security, Volume-1, Issue-1,2007.

[3] Dr. Mohammed M. Alani, "DES96- Improved DES Security" in Proeceding of 7th International Multi-Conference on System, Signals and Devices, 2010.

[4] N.Bhaskar, "Symmetric Key Cryptography Algorithm Using Complement For Small Data Security" In the International Journal of Engineering Research & Technology, Volume-2, Issue-5, May-2013.

[5] Bibhudendra Acharya, Sarat Kumar Patra, Ganapati Panda, "Involutory, Permuted and Reiterative Key Matrix Generation Methods for Hill Cipher System" In the International Journal of Recent Trend in Engineering, Volume-1, No. 4, May 2009.

[6] M. Nordin A. Rahman, A.F.A.Abidin, Terengganu, "Cryptography: A New Approach of Classical Hill Cipher" In the International Journal of Security and Its Application, Volume-7, No.2, March-2013.

[7] Richard Clayton and Mike Bond, "Experience Using a Low-Cost FPGA Design to Crack DES Keys" Ches 2002.

[8] MD Asif Mushtaque, Harsh Dhiman, Shahnawaz Hussain, Shivangi Maheshwari, "Evaluation of DES, TDES, AES, Blowfish and Two fish Encryption Algorithm: Based on Space Complexity" In the International Journal of Engineering & Technology, Volume-3, Issue-4, April-2014.

[9] Dr. Atul M. Gonsai, Lakshadeep M. Raval, "Evaluation of Common Encryption Algorithms and Scope of Advanced Algorithms for Simulated Wireless Network" In the International Journal of Computer Trends and Technology, Volume-11, Number 1, May-2014.
[10] Horst Feistel,"Cryptography and Computer Privacy" In the Scientific American, Volume-228, No.5, May-1973.
[11] William Stallings, "Hill Cipher and Modular Arithmetic & Data Encryption Standards" In the
[12] Cryptography and Network Security Principles and Practice Book, Fifth Edition.
[13] Arvind Kumar Sharma, Hitesh Sharma, "A Survey on Common Encryption Algorithms" In the International Journal of Engineering Research & Technology, Volume-3, Issues-12, December-2014.
[14] J. Orlin Grabbe, "The DES Algorithms Illustrated" In the Laissez Faire City Times, Volume-2, No.-28.

AUTHOR PROFILE

Arvind Kumar Sharma is student of Master of Technology in Computer Science and Engineering at Lovely Professional University, Phagwara, Punjab. He has received hism MSc.Computer Science degree from Doaba College, Jalandhar in 2011. His current area of Interest in research is "Networking and Security" and in Technology is "Programming".